METHOD AND SYSTEM FOR MULTIFORMAT PRESENTATION

BACKGROUND OF THE INVENTION

1. Technical Field:

The present invention relates in general to data processing systems and in particular to stored documents within data processing systems. More particularly, the present invention relates presentation of stored documents to various output peripherals of the data processing system. Even more particularly the present invention relates to efficiency in document presentation to the output peripherals.

2. Description of the Related Art:

Presentation of stored data within a data processing system may take the form of pixels on a display or dots printed by a printer. Each output mechanism has different requirements for presentation. Each mechanism typically requires different presentation parameters, such as the color space and resolution. For example a screen will commonly require 24 bit RGB data at 72 DPI, while a printer might require 32 bit CMYK data at 600 DPI.

"Tiles" are utilized to break objects or data into multiformat units and use various tile encoding schemes based on the data within a tile (e.g., line art image compressed with G4 coding but a contone image compressed with JPEG). Some systems have also used tiles to describe each object on a page using, at most, one tile per object. Usually the tiles do not overlap, but some systems allow the overlapping of tiles.

Currently, images stored for presentation on different devices within the same system may use OPI (Open PrePress Interface) technology or FlashPix technology. OPI is

primarily utilized in printing and FlashPIX, jointly developed by Kodak™, Hewlett-Packard Company™, Live Picture, Inc.™ and Microsoft™, is a multi-resolution, tiled file format that stores images at different resolutions. In both cases images are stored in full (i.e., not broken into tiles and each tile stored separately) and the main purpose is to store the images in different resolutions. Since the images are not divided into units, and non-image data is not supported, both FlashPix and OPI technology are generally limited to image files.

Processing power needs to be invoked in an output device (which may not have much processing power) at presentation time. Utilizing a presentation device efficiently requires adequate processing power for each job sent to the device. Accordingly, what is needed is a system and method that allows stored documents to be presented on a variety of output devices with a reduced requirement for processing resources. Additionally, what is needed is a method and system that reduces presentation time of stored documents. Further, what is needed is a system and method that provides a more efficient use of computing resources.

SUMMARY OF THE INVENTION

It is therefore one object of the present invention to provide a method and system that will reduce the time between a document request and presentation of the document.

It is another object of the present invention to provide a method and system for providing a more efficient use of processing power in a data processing system.

It is yet another object of the present invention to provide a method and apparatus for maximizing the achievable output quality when output objects are stored in a persistent storage.

The foregoing objects are achieved as is now described. A method and system for storing data in multiple formats based on the nature of the data and the characteristics of the possible output devices are disclosed. The disclosed invention, when coupled with a data processing system and multiple presentation devices, is designed to minimize processing requirements and processing time while maximizing output quality. The data processing system includes a central processing unit, memory, at least one output device, and a user input device. The method and system provide outputting of a data set to a physical output device such as a display, a printer, a fax, or a logical output device such as an email generator or any other data processing system. In implementing the improved method and system for improving the data set presentation process a data set is broken into objects and further into units so that each unit within an object contains a similar data type. Units that require less processing power for presentation are stored in a device-independent format. Units that require more processing power for presentation are stored in device-independent format and device dependent format determined by the presentation parameters of an attached peripheral

presentation device. At presentation time a document database, or storage area, assembles the document from the units determined by the presentation device. The Document is composed of data that is specific for the presentation device or data that is device independent.

The above as well as additional objects, features, and advantages of the present invention will become apparent in the following detailed written description.

BRIEF DESCRIPTION OF THE DRAWINGS

The novel features believed characteristic of the invention are set forth in the appended claims. The invention itself however, as well as a preferred mode of use, further objects and advantages thereof, will best be understood by reference to the following detailed description of an illustrative embodiment when read in conjunction with the accompanying drawings, wherein:

Figure 1 depicts a high-level block diagram of a data processing system in which a preferred embodiment of the present invention may be implemented;

Figure 2 depicts a high-level block diagram of the data processing system utilizing a document process in accordance with a preferred embodiment of the present invention; and

Figure 3 illustrates a high-level flow diagram of a process for improving processing requirements and processing time for document presentation in accordance with a preferred embodiment of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

Referring now to the figures, and in particular to **Figure 1**, a network data processing system, in which the present invention can be employed is depicted. As shown data processing system **100** comprises a number of components, which are interconnected together through a basic network. Server **102**, computer system **105**, printer **110**, display **104** are connected to form a rudimentary network wherein server **102** serves printer **110** and computer system **105**. More particularly, computer system **105** is coupled to and can drive optional monitor **104** (such as a conventional video display). Computer system **105** can be optionally coupled to input devices such as keyboard **106** or mouse **108**. Mouse **108** includes right and left buttons (not shown). An optional output device, such as printer **110**, also can be connected to computer system **105**. Finally, computer system **105** may include one or more mass storage devices such as diskette drive **112**.

Computer system **105** responds to input devices, such as keyboard **106**, mouse **108**, or local area networking interfaces. Additionally, input/output (I/O) devices, such as floppy diskette drive **112**, display **104**, printer **110**, and local area network communication system (not shown) are connected to computer system **105** in a manner well known. Of course, those skilled in the art are aware that other conventional components also can be connected to the computer system **105** for interaction therewith. In accordance with the present invention, computer system **105** includes a system processor that is interconnected to a

random access memory (RAM), a read only memory (ROM), and a plurality of I/O devices.

Specifically, computer system **105** may be implemented utilizing any suitable computer such as the IBM PS/2 computer or an IBM RISC SYSTEM/6000 computer, both products of International Business Machines Corporation, located in Armonk, N.Y. "RISC SYSTEM/6000" is a trademark of International Business Machines Corporation (IBM) and "PS/2" is a registered trademark of IBM.

Referring to **Figure 2**, a high-level block diagram of the data processing system utilizing a document process in accordance with a preferred embodiment of the present invention, is depicted. Document **202** is scanned and separated into objects by de-composer **211**, as is common in modern datastream architectures, such as Mixed Object: Document Content Architecture (MO:DCA™), a product of IBM. The complete document is broken down into objects with fixed **204** representing a block of text on the document, pie chart **206** representing a vector graphics object and object **207** representing and arbitrary object that is composed of tiles **208 - 210**. For this example, the document is stored in CIELab format. CIELab color gets its name from a color space that uses three values to describe the precise three-dimensional location of a color inside a visible color space. CIE stands for Commission Internationale de l-Eclairages an international body of color scientists whose standards make it possible to communicate color information accurately. **L** describes relative lightness; **A** represents relative redness-greenness, and **B** represents relative yellowness-blueness.

Each tile in the source document is preferably specified in device-independent format. For example, if the tiles are part of an image object, the color is specified in a device-independent color space such as CIELab. However, to optimize presentation speed, each tile is regenerated in a number of device-specific formats that are tuned to the various output devices on the system. A system that is required to present documents on display screen **222**, print documents on a full-color CMYK printer and proof documents on a monochrome printer might generate every color image tile in three additional device-dependent formats: 24 bit RGB at 72 dots per inch for the display; 32 bit CMYK at 300 dpi for the color printer and 1 bit bi-level at 600 dpi for the monochrome proofing printer. This relieves the output device from having to spend processing power and time to convert the tiles to device-specific format.

Object **207** is divided into tiles that become storage units. Each tile is encoded using the device independent format (e.g.,CIELab data) and in device specific formats. A system that is required to present documents on display screen **222**, print the documents on a 32 bit CMYK printer or on a bilevel proof printer might keep every color image tile in four formats: 24 bit CIELab, 24 bit RGB, 32 bit CMYK and bilevel. CMYK and bilevel tiles would be generated for the particular printers used, since both conversion to CMYK and halftoning to bilevel are heavily device dependent.

Document **202** is broken down into objects for storage and transmitted to a storage medium, in this instance a database residing on a server in the system. Transformers **214** is logic on the system that stores the objects in the proper and efficient format. Like objects (common format) are stored with like objects in the database. At

presentation time, server **216** receives a request to print document **202** to a particular printer (printer 1 **218**) and at the same time display document **202** on display **222**. If Printer 1 **218** is a full-color, 300 dpi CMYK printer, any objects or tiles that require intensive computation to convert from device-independent format, such as CIELab raster image tiles, are sent to the printer in the previously generated device-dependent format. Objects that are transformed easily into printer format, such as color text, are sent to the printer in device-independent format. Similar processing occurs for the display, except in this case, the device-dependent format for raster image tiles is 24 bit RGB at 72 dpi. Database **212** assembles the document from the stored units (tiles) based on the requirements of the presentation device (printer and display). The document received by the particular device is composed of data that is either specific for that device or device independent, but easy to convert to device specific form.

Referring to **Figure 3**, a high-level flow diagram of a process for improving processing requirements and processing time for document presentation in accordance with a preferred embodiment of the present invention, is illustrated. The process is divided into two stages: a first stage that processes and stores each document in the system (comprising steps **300** - **308**) and a second stage that serves the stored document to receivers such as printers and displays (comprising steps **309** - **318**).

The process begins with step **300**, which depicts analyzing a document received by the system to determine possible recipient devices present and available on the system. It is important to note that possible output

devices may vary from document to document. The process proceeds to step **301**, which illustrates analyzing the format of the document. The step then passes to step **302**, which depicts parsing the document into objects. Next, the process moves to step **303**, which illustrates dividing each object into component units. For example, if the object is an image each data unit could be a tile.

The process continues with step **304**, which depicts storing the component units in device-independent format to preserve unit information. The process next proceeds to step **305**, which illustrates a determination of whether the possible recipients of the document are known devices. If the possible recipients of the document are not known devices, the process passes to step **309**, terminating the first stage. If the recipients of the document are known devices the process instead proceeds to step **306**, which depicts a determination of whether a priori mode should be utilized. In a priori mode, the device-dependent formats are computed in advance for selected known output devices (recipients). If the determination is made that the a priori mode is not utilized, the process passes to step **309**, terminating the first stage. If the determination is made that the a priori mode is utilized, the process instead proceeds to step **307**, which depicts a determination of whether the unit data type is complex.

If the determination is made that the unit data type is not complex, the process proceeds to step **309**, which illustrates beginning the second stage. Units classified as "not complex" are data types that do not require intensive processing for generation of the device-specific format. An example would be text designed to be printed as black. If

the determination is made that the unit data type is complex, the process then proceeds to step **308**, which depicts generating and storing device-dependent formats for each selected device in the system. Note that selected devices may vary for different data types in the same system since a data type may be complex for some device types and not complex for others. The range of devices is a subset of the devices determined to be on the system in step **300**.

The process continues with step **309**, which illustrates receiving a request for a particular document. The process then proceeds to step **310**, which depicts a determination of whether the targeted recipient device is known. The device is "known" if its device-dependent data formats are known. If the determination is made that the device is not known, the process passes to step **317**, which illustrates sending the unit data in device independent format to a receiving device such as a printer or display. The process continues to step **318**, which depicts the receiving device receiving and processing the data.

Returning to step **310**, if the determination is made that the device is known, the process instead passes to step **311**, which illustrates determining whether the data is a complex data unit. If the determination is made that the unit of data is not complex for the target output device, the process passes to step **317**, which illustrates sending the unit data in device independent format to a receiving device such as a printer or display. The process continues to step **318**, which depicts the receiving device receiving and processing the data. If the determination is made that the unit is complex for the target device, the process continues to step **312**, which illustrates a determination of

whether the unit is stored in device-dependent format for the target device. If the determination is made that the unit is stored in device-dependent format, the process proceeds to step **316**, which depicts sending the unit to the target device in device-dependent format. If the determination is made that the unit is stored in device-independent rather than device-dependent format, the process proceeds instead to step **313**, which illustrates the system generating device-dependent format from the device-independent format. As detailed above, device-independent format is always available as it has been stored in that format to preserve all the information (see step **304**).

The process then passes to step **314**, which depicts a determination of whether the data should be saved for later re-use. Different rules may be formulated to determine caching such as a likelihood that this unit will be requested again by the target receiving device or the storage capacity, usage and processing required to generate the device-dependent format. If the determination is made that the device-dependent format is not to be cached, the process passes to step **316**, which illustrates sending the unit in the device dependent format to the target receiving device. If the determination is made that the device-dependent unit is to be cached, the process instead moves to step **315**, which depicts the unit being stored in device-dependent format. The process then proceeds to step **316**, which illustrates sending the unit, in device-dependent format, to the target receiving device. The process then passes to step **318** which depicts the target receiving device receiving and processing the unit data for presentation

Once the document or data set is divided into units, the question is how much processing power is required to convert a unit (of the document) from device-independent format to device-dependent format. The present invention precludes the need to invoke significant processing power in the presentation (output) device at presentation time. The present invention invokes the processing power in the server before presentation time.

This invention is described in a preferred embodiment in the following description with reference to the figures, in which like numbers represent the same or similar elements. While this invention is described in terms of the best mode for achieving this invention's objectives, it will be appreciated by those skilled in the art that variations may be accomplished in view of these teachings without deviating from the spirit or scope of the present invention. For example, the present invention may be implemented using any combination of computer programming software, firmware or hardware. As a preparatory step to practicing the invention or constructing an apparatus according to the invention, the computer programing code (whether software or firmware) according to the invention will typically be stored in one or more machine readable storage mediums such as fixed (hard) drives, diskettes, optical disks, magnetic tape, semiconductor memories such as ROMs, PROMs, etc., thereby making an article of manufacture in accordance with the invention. The article of manufacture containing the computer programming code is used by either executing the code directly from the storage device, by copying the code from the storage device into another storage device such as a hard disk, RAM, etc. or by transmitting the code on a network for remote execution. The method form of the invention may be practiced by combining one or more machine

readable storage devices containing the code according to the present invention with appropriate standard computer hardware to execute the code contained therein. An apparatus for practicing the invention could be one or more computers and storage systems containing or having network access to computer program(s) coded in accordance with the invention.

While the invention has been particularly shown and described with reference to a preferred embodiment, it will be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention.